Git is a distributed version control system.  
Staging area – files that will be tracked by git.

* Git config --global user.name “Manideepa Shaw”
* Git config --global.email “jiya@gmail.com”

1. **Git init***- initializes the current folder as git repository*
2. **Git init –b main**

* to change the main branch name from master to main

1. **Git add .  
   -** *Adds all the files*
2. **Git commit***- commits changes to the git repository*
3. **Git commit –a –m “Committed”**

* *When I don’t want to do git add separately*

1. **Git push (for an already existing file)/** for a new file don’t add readme file
2. Git status  
   - Gives the status of the repository(if any commit is not done yet,etc.)
3. Git log/ git log --oneline  
   - shows all the previous commits
4. git reset HEAD gradientcolours.docx  
   git checkout -- gradientcolours.docx
5. git clone repository\_url  
   - for copying a repository from an url to our machine
6. **adding to git repo**  
   git remote add origin <https://github.com/manideepa-shaw/MoResins.git>

git branch -M main

git push -u origin main

1. G**it diff**

* Shows what changes have been made

**OR**

1. **Git diff --staged**

* If we have done git add . (i.e., moved the files from working directory to staged one)

1. Git rm --cached *filename //To remove the file after being commited, i.e., to untrack the file*
2. Git remote –v //*to know the remote - > in this case origin*
3. Git push origin main //*to push to a specific branch*
4. Git tag –a v1.0 –m “version 1” //*used for defining versions*
5. Git tag //*shows all the versions*
6. Git push origin v1.0 //*pushes the tag to the github*
7. Git stash //*when we have done some changes but now I want to go back to the previous commit but also I don’t want to commit the current changes nor loose them, then we can do this*
8. git stash list //*to list all the stash*
9. git stash apply //*to go back to the changed stuffs*

**Branches**

1. git checkout -b file1 //***creation*** *of branch(file1 is the branch name)*
2. git checkout file1 //***switching*** *to another branch*
3. git switch - // *switches to the branch from where it was created*
4. git branch –d file1 //deleting the branch
5. git push origin --delete <branch-name> //*to delete branch remotely also*
6. git push origin f1 // pushing the branch to the server
7. git merge <*branchName*> //*ensure to remain on the branch from where we made the branch. It always better to do first :* ***git pull origin main***
8. git rebase branchname //*does same as branch but doesnot shows the branches once merged*

**Git Time travel(switches between different commits)**

1. git checkout <commit number> //*goes to the previous commit*
2. git checkout –b *<newBranch*>  *//new branch will be made at this from this commit*
3. \*\*not sure  
   git checkout <commit number> filename  
   - goes to the previous commit(if u want this one, again make a commit or if u want to go back to your final commit follow (this same step again) )

**If .git file gets deleted by mistake from the folder**

1. git init
2. git remote add origin <https://github.com/manideepa-shaw/MoResins.git>
3. Move all the files of the current folder in a new folder (say backup)
4. git pull origin main
5. Now do whatever you want to, delete the files, move file to the required folder, etc then commit and push